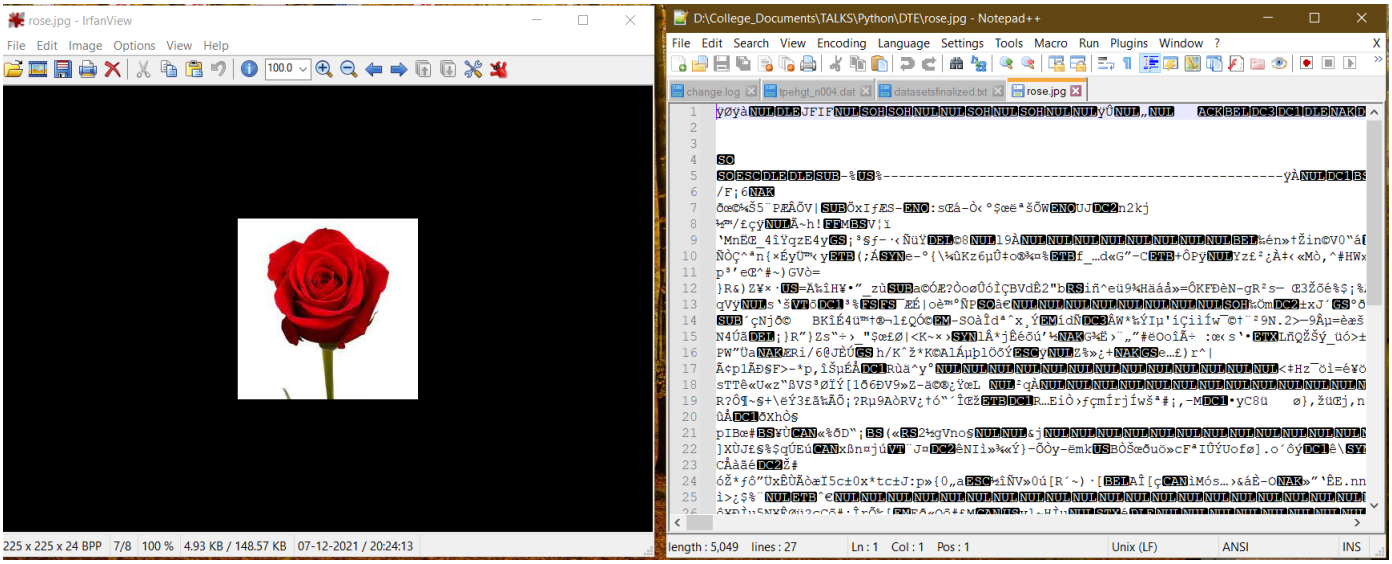
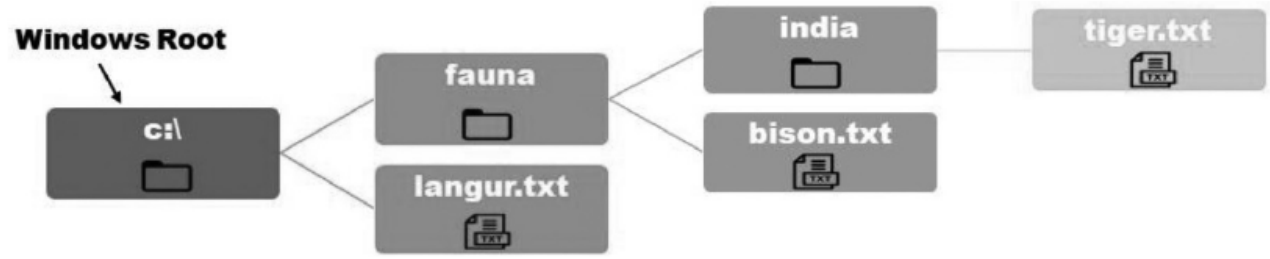
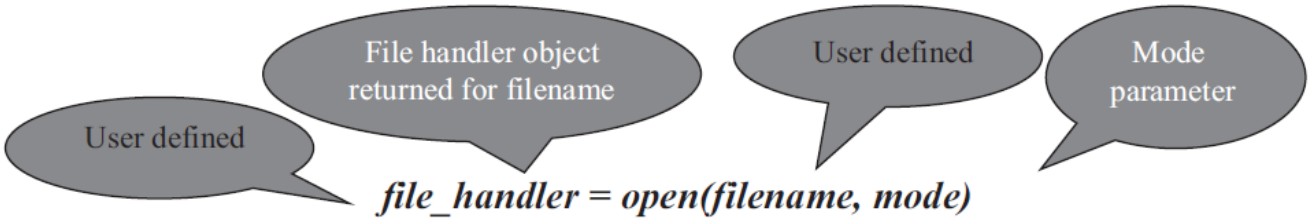
Types of Files

1. Binary Files
2. Text Files



#Fully Qualified Path and Realtive path "C:\langur.txt" --> Fully Qualified Path "..\langur.txt" --> Relative Path





Access Modes of the Files

Mode	Description
"r"	Opens the file in read only mode and this is the default mode.
"w"	Opens the file for writing. If a file already exists, then it'll get overwritten. If the file does not exist, then it creates a new file.
"a"	Opens the file for appending data at the end of the file automatically. If the file does not exist it creates a new file.
"r+"	Opens the file for both reading and writing.
"w+"	Opens the file for reading and writing. If the file does not exist it creates a new file. If a file already exists then it will get overwritten.
"a+"	Opens the file for reading and appending. If a file already exists, the data is appended. If the file does not exist it creates a new file.
"x"	Creates a new file. If the file already exists, the operation fails.
"rb"	Opens the binary file in read-only mode.
"wb"	Opens the file for writing the data in binary format.
"rb+"	Opens the file for both reading and writing in binary format.

In [2]:

```
#creates the file example.txt if it is not present already
file_handler = open("example.txt","x")
```

In [4]:

```
#opens the file moon.txt which is present already
file_handler = open("moon.txt","r")
```

In [6]:

```
#opens the file named "langur.txt" which is present in C directory
file_handler = open("C:\langur.txt","r")
```

In [7]:

```
file_handler = open(r"D:\network\computer.txt","r")
```

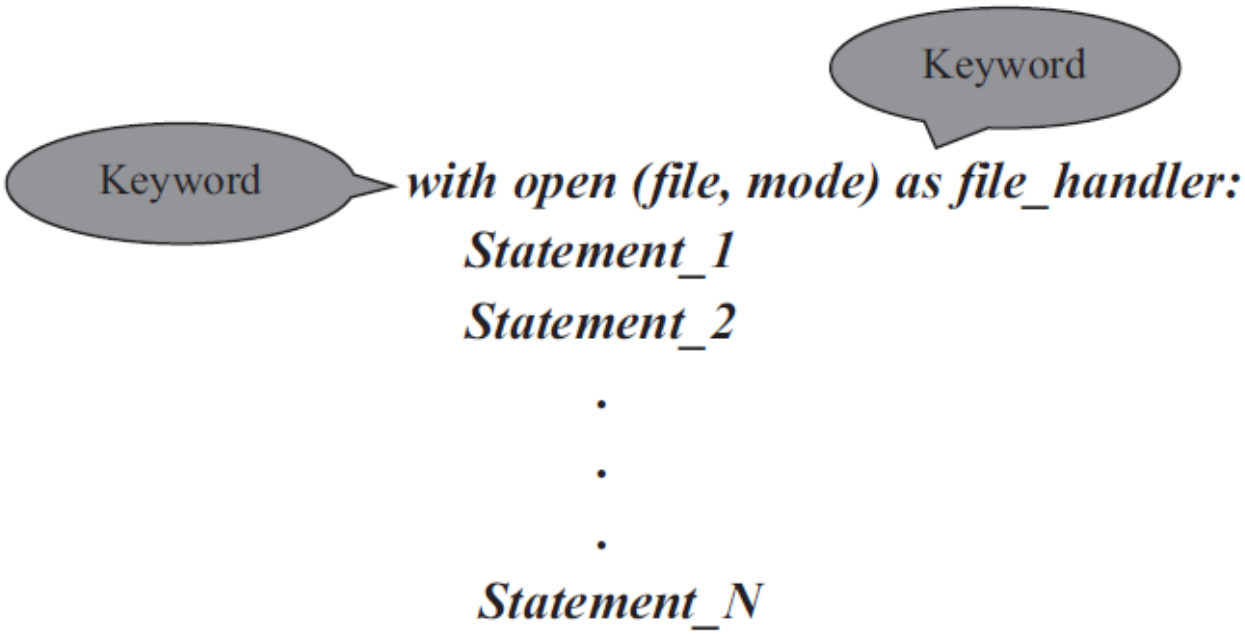
The syntax for close() function is `file_handler.close()`

#Syntax for try-catch-except finally block try: f = open("file", "w") try: f.write('Hello World!') finally: f.close() except IOError: print('oops!')

In [8]:

```
# Program 9.1: Write Python Program to Read and Print Each Line in "egypt. txt" file.
# Sample Content of "egypt.txt" File is Given Below.

def read_file():
    file_handler = open("egypt.txt")
    print("Printing each line in text file")
    for each_line in file_handler:
        print(each_line)
    file_handler.close()

def main():
    read_file()

if __name__ == "__main__":
    main()
```

```
Printing each line in text file
Ancient Egypt was an ancient civilization of eastern North Africa, concentra
ted along the lower reaches of the Nile River.

The civilization coalesced around 3150 BC with the political unification of
Upper and Lower Egypt under the first pharaoh.

Ancient Egypt reached its pinnacle during the New Kingdom, after which it en
tered a period of slow decline.
```



In [66]:

```
# Program 9.2: Program to Read and Print Each Line in "japan.txt" File Using
# with Statement. Sample Content of "japan.txt" File is Given Below.

def read_file():
    print("Printing each line in text file")
    with open("japan.txt") as file_handler:
        for each_line in file_handler:
            print(each_line.strip())

def main():
    read_file()

if __name__ == "__main__":
    main()
```

```
Printing each line in text file
National Treasures of Japan are the most precious of Japan's Tangible Cultur
al Properties.
A Tangible Cultural Property is considered to be of historic or artistic val
ue, classified either as
"buildings and structures", or as "fine arts and crafts".
```

List of File Attributes

Attribute	Description
file_handler.closed	It returns a Boolean True if the file is closed or False otherwise.
file_handler.mode	It returns the access mode with which the file was opened.
file_handler.name	It returns the name of the file.

In [12]:

```
file_handler = open("computer.txt", "w")
print(f"File Name is {file_handler.name}")
print(f"File State is {file_handler.closed}")
print(f"File Opening Mode is {file_handler.mode}")
```

```
File Name is computer.txt
File State is False
File Opening Mode is w
```

List of Methods Associated with the File Object

Method	Syntax	Description
read()	file_handler. read([size])	This method is used to read the contents of a file up to a size and return it as a string. The argument *size* is optional, and, if it is not specified, then the entire contents of the file will be read and returned.
readline()	file_handler.readline()	This method is used to read a single line in file.
readlines()	file_handler.readlines()	This method is used to read all the lines of a file as list items.
write()	file_handler. write(string)	This method will write the contents of the string to the file, returning the number of characters written. If you want to start a new line, you must include the new line character.

List of Methods Associated with File Object

Method	Syntax	Description
writelines()	file_handler. writelines(sequence)	This method will write a sequence of strings to the file.
tell()	file_handler.tell()	This method returns an integer giving the file handler's current position within the file, measured in bytes from the beginning of the file.
seek()	file_handler. seek(offset, from_what)	This method is used to change the file handler's position. The position is computed from adding *offset* to a reference point. The reference point is selected by the *from_what* argument. A *from_what* value of 0 measures from the beginning of the file, 1 uses the current file position, and 2 uses the end of the file as the reference point. If the *from_what* argument is omitted, then a default value of 0 is used, indicating that the beginning of the file itself is the reference point.

In [2]:

```
f = open("example.txt", "w")
f.write("Thinking Python Programming")
f.close()
```

In [3]:

```
f = open("example.txt")
print(f.read(2))
print(f.read(2))
print(f.read(2))
print(f.read(2))
```

```
Th
in
ki
ng
```

In [67]:

```
# Program 9.3: Write Python Program to Read "rome.txt" File Using
# read() Method. Sample Content of "rome.txt" File is Given Below

def main():
    with open("rome.txt") as file_handler:
        print("Print entire file contents")
        print(file_handler.read(), end="")
        #print(file_handler.read(13), end="")

if __name__ == "__main__":
    main()
```

```
Print entire file contents
Ancient Rome was a civilization which began on the Italian Peninsula in the
8th century BC.
The Roman Emperors were monarchial rulers of the Roman State.
The Emperor was supreme ruler of Rome.
Rome remained a republic.
```

In [68]:

```
# Program 9.4: Consider the "rome.txt" File Specified in Program 9.3. Write
# Python Program to Read "rome.txt" file Using readline() Method

def main():
    with open("rome.txt") as file_handler:
        print("Print a single line from the file")
        print(file_handler.readline(), end="")
        print("Print another single line from the file")
        print(file_handler.readline(), end="")

if __name__ == "__main__":
    main()
```

```
Print a single line from the file
Ancient Rome was a civilization which began on the Italian Peninsula in the
8th century BC.
Print another single line from the file
The Roman Emperors were monarchial rulers of the Roman State.
```

In [15]:

```
# Program 9.5: Consider the "rome.txt" File Specified in Program 9.3. Write
# Python Program to Read "rome.txt" File Using readlines() Method

def main():
    with open("rome.txt") as file_handler:
        print("Print file contents as a list")
        print(file_handler.readlines())

if __name__ == "__main__":
    main()
```

```
Print file contents as a list
['Ancient Rome was a civilization which began on the Italian Peninsula in th
e 8th century BC.\n', 'The Roman Emperors were monarchial rulers of the Roma
n State.\n', 'The Emperor was supreme ruler of Rome.\n', 'Rome remained a re
public.']
```

In [16]:

```
file_handler = open("moon.txt","w")
file_handler.write("Moon is a natural satellite")
file_handler.close()
```

In [17]:

```
file_handler = open("moon.txt", "a+")
file_handler.write("of the earth")
file_handler.close()
```

In [18]:

```
file_handler = open("moon.txt")
file_handler.read()
```

Out[18]:

```
'Moon is a natural satelliteof the earth'
```

In [20]:

```
file_handler.close()
file_handler = open("moon.txt","w")
file_handler.writelines(["Moon is a natural satellite", " ", "of the earth"])
file_handler.close()
```

In [21]:

```
file_handler = open("moon.txt")
file_handler.read()
```

Out[21]:

```
'Moon is a natural satellite of the earth'
```

In [22]:

```
file_handler.close()
```

In [4]:

```
f = open('workfile', 'w')
f.write('yy1234568888')
f.close()
```

In [5]:

```
f = open('workfile', 'r')
print(f.seek(5))
print(f.read())
```

```
5
4568888
```

In [27]:

```
f = open('workfile', 'w')
f.write('0123456789abcdef')
```

Out[27]:

```
16
```

In [28]:

```
f.close()
```

In [29]:

```
f = open('workfile', 'rb+')
f.seek(2)
```

Out[29]:

```
2
```

In [30]:

```
f.seek(2, 1)
```

Out[30]:

```
4
```

In [31]:

```
f.read()
```

Out[31]:

```
b'456789abcdef'
```

In [32]:

```
f.seek(-3, 2)
```

Out[32]:

```
13
```

In [33]:

```
f.read()
```

Out[33]:

```
b'def'
```

In [34]:

```
f.tell()
```

Out[34]:

```
16
```

In [7]:

```
# Program 9.6: Consider "Sample_Program.py" Python file. Write Python program
# to remove the comment character from all the lines in a given Python source
# file. Sample content of "Sample_Program.py" Python file is given below

def main():
    with open("Sample_Program.py") as file_handler:
        for each_row in file_handler:
            each_row = each_row.replace("#", "")
            print(each_row, end="")

if __name__ == "__main__":
    main()
```

```
print("This is a sample program")
print("Python is a very versatile language")
```

In [71]:

```
# Program 9.7: Write Python Program to Reverse Each Word in "secret_societies.txt" file.
# Sample Content of "secret_societies.txt" is Given Below

def main():
    reversed_word_list = []
    with open("secret_societies.txt") as file_handler:
        for each_row in file_handler:
            word_list = each_row.rstrip().split(" ")
            for each_word in word_list:
                reversed_word_list.append(each_word[::-1])
            print(" ".join(reve
            rsed_word_list))
            reversed_word_list.clear()

if __name__ == "__main__":
    main()
```

```
terceS seiteicoS
snosameerF itanimullI
snaicurcisoR grebredliB sthginK ralpmeT
```

In [72]:

```
# Program 9.8: Write Python Program to Count the Occurrences of Each Word
# and Also Count the Number of Words in a "quotes.txt" File. Sample
# Content of "quotes.txt" File is Given Below

def main():
    occurrence_of_words = dict()
    total_words = 0
    with open("quotes.txt") as file_handler:
        for each_row in file_handler:
            words = each_row.rstrip().split()
            total_words += len(words)
            for each_word in words:
                occurrence_of_words[each_word] = occurrence_of_words.get(each_word, 0) + 1
        print("The number of times each word appears in a sentence is")
        print(occurrence_of_words)
        print(f"Total number of words in the file are {total_words}")

if __name__ == "__main__":
    main()
```

```
The number of times each word appears in a sentence is
{'Happiness': 1, 'is': 2, 'the': 1, 'longing': 1, 'for': 2, 'repetition.':
1, 'Artificial': 1, 'intelligence': 1, 'no': 1, 'match': 1, 'natural': 1, 's
tupidity.': 1}
Total number of words in the file are 14
```

In [73]:

```
# Program 9.9: Write Python Program to Find the Longest Word in a File. Get the File
# Name from User. (Assume User Enters the File Name as "animals.txt" and its
# Sample Contents are as Below)

def read_file(file_name):
    with open(file_name) as file_handler:
        longest_word = ""
        for each_row in file_handler:
            word_list = each_row.rstrip().split()
            for each_word in word_list:
                if len(each_word) > len(longest_word):
                    longest_word = each_word
    print(f"The longest word in the file is {longest_word}")

def main():
    file_name = input("Enter file name: ")
    read_file(file_name)

if __name__ == "__main__":
    main()
```

```
Enter file name: rome.txt
The longest word in the file is civilization
```

In [74]:

```
# Program 9.10: Write Python Program to Create a New Image from an Existing Image

def main():
    with open("rose.jpg", "rb") as existing_image, open("new_rose.jpg", "wb") as new_image:
        for each_line_bytes in existing_image:
            new_image.write(each_line_bytes)

if __name__ == "__main__":
    main()
```

In [75]:

```
# Program 9.11: Consider a File Called "workfile". Write Python Program to Read and
# Print Each Byte in the Binary File

def main():
    with open("workfile", "wb") as f:
        f.write(b'abcdef')

    with open("workfile", "rb") as f:
        byte = f.read(1)
        print("Print each byte in the file")
        while byte:
            print(byte)
            byte = f.read(1)

if __name__ == "__main__":
    main()
```

```
Print each byte in the file
b'a'
b'b'
b'c'
b'd'
b'e'
b'f'
```

In [36]:

```
#Understand bytes in detail
print(b'Hello')
type(b'Hello')
```

```
b'Hello'
```

Out[36]:

```
bytes
```

In [37]:

```
for i in b'Hello':
    print(i)
```

```
72
101
108
108
111
```

In [38]:

```
bytes(3)
```

Out[38]:

```
b'\x00\x00\x00'
```

In [39]:

```
bytes([70])
```

Out[39]:

```
b'F'
```

In [40]:

```
bytes([72, 101, 108, 108, 111])
```

Out[40]:

```
b'Hello'
```

In [42]:

```
print(b'\x61')
bytes('Hi', 'utf-8')
```

```
b'a'
```

Out[42]:

```
b'Hi'
```

The syntax for bytes() class method is, bytes(source[, encoding])

In [76]:

```
# Program 9.12: Write Python Program to Save Dictionary in Python Pickle

import pickle

def main():
    bbt = {'cooper': 'sheldon'}
    with open('filename.pickle', 'wb') as handle:
        pickle.dump(bbt, handle)
    with open('filename.pickle', 'rb') as handle:
        bbt = pickle.load(handle)
        print(f"Unpickling {bbt}")

if __name__ == "__main__":
    main()
```

```
Unpickling {'cooper': 'sheldon'}
```

```
The syntax used to read from a CSV file is to use csv.reader() method.
The syntax used is
    csv.reader(csvfile)
```

In [77]:

```
# Program 9.13: Write Python program to read and display each row in
# "biostats.csv" CSV file. Sample content of "biostats.csv" is given below.

import csv

def main():
    with open('biostats.csv', newline='') as csvfile:
        csv_reader = csv.reader(csvfile)
        print("Print each row in CSV file")
        for each_row in csv_reader:
            #print(f'{",".join(each_row)}')
            print(",".join(each_row))

if __name__ == "__main__":
    main()
```

```
Print each row in CSV file
Name,     "Sex", "Age", "Height (in)", "Weight (lbs)"
Alex,       "M",   41,       74,      170
Bert,       "M",   42,       68,      166
Carl,       "M",   32,       70,      155
Dave,       "M",   39,       72,      167
Elly,       "F",   30,       66,      124
Fran,       "F",   33,       66,      115
Gwen,       "F",   26,       64,      121
Hank,       "M",   30,       71,      158
Ivan,       "M",   53,       72,      175
Jake,       "M",   32,       69,      143
Kate,       "F",   47,       69,      139
Luke,       "M",   34,       72,      163
Myra,       "F",   23,       62,       98
Neil,       "M",   36,       75,      160
Omar,       "M",   38,       70,      145
Page,       "F",   31,       67,      135
Quin,       "M",   29,       71,      176
Ruth,       "F",   28,       65,      131
```

In [78]:

```
# Program 9.14: Write Python program to read and display rows in "employees.csv" CSV file
# that start with employee name "Jerry". Sample content of "employees.csv" is given below

import csv

def main():
    with open('employees.csv', newline='') as csvfile:
        csv_reader = csv.reader(csvfile)
        print("Print rows in CSV file that start with employee name 'Jerry'")
        for each_row in csv_reader:
            if each_row[0] == "Jerry":
                print(",".join(each_row))

if __name__ == "__main__":
    main()
```

```
Print rows in CSV file that start with employee name 'Jerry'
Jerry,Male,01-10-2004,12:56 PM,95734,19.096,FALSE,Client Services
Jerry,Male,03-04-2005,1:00 PM,138705,9.34,TRUE,Finance
```

The syntax for write method in csv writer, csv.writer(csvfile) csvwriter.writerow(row) csvwriter.writerows(rows)

In [8]:

```
# Program 9.15: Write Python program to write the data given below to a CSV file.
# Category,Winner,Film,Year
# Best Picture,Doug Mitchell and George Miller,Mad Max: Fury Road,2015
# Visual Effects,Richard Stammers,X-Men:Days of Future Past,2014
# Best Picture,Martin Scorsese and Leonardo DiCaprio,The Wolf of Wall Street,2013
# Music(Original Song),Adele Adkins and Paul Epworth,Skyfall from Skyfall,2012

import csv

def main():
    csv_header_name = ['Category', 'Winner', 'Film', 'Year']
    each_row = [['Best Picture', 'Doug Mitchell and George Miller', 'Mad Max: Fury Road', '
                 ['Visual Effects', 'Richard Stammers', 'X - Men: Days of Future Past', '201
                 ['Best Picture', 'Martin Scorsese and Leonardo DiCaprio', 'The Wolf of Wall
                 ['Music(Original Song)', 'Adele Adkins and Paul Epworth', 'Skyfall from Sky

    with open('oscars.csv', 'w', newline='') as csvfile:
        csv_writer = csv.writer(csvfile)
        csv_writer.writerow(csv_header_name)
        csv_writer.writerows(each_row)

if __name__ == "__main__":
    main()
```

The syntax for DictReader and DictWriter is **class csv.DictReader(f, fieldnames=None, restkey = None) ** **class csv.DictWriter(f, fieldnames, extrasaction='raise') **

In [1]:

```
# Program 9.16: Write Python Program to Read Data from ''pokemon.csv'' csv
# File Using DictReader. Sample Content of ''pokemon.csv'' is Given Below

import csv

def main():
    with open('pokemon.csv', newline='') as csvfile:
        reader = csv.DictReader(csvfile)
        for row in reader:
            print(f"{row['Pokemon']}, {row['Type']}")

if __name__ == "__main__":
    main()
```

```
Bulbasaur, Grass
Charizard, Fire
Squirtle, Water
Pikachu, Electric
Rapidash, Fire
```

In [44]:

```
# Program 9.17: Write Python program to demonstrate the writing of data
# to a CSV file using DictWriter class
import csv
def main():
    with open('names.csv', 'w', newline='') as csvfile:
        field_names = ['first_name', 'last_name']
        writer = csv.DictWriter(csvfile, fieldnames=field_names)
        writer.writeheader()
        writer.writerow({'first_name': 'Baked', 'last_name': 'Beans'})
        writer.writerow({'first_name': 'Lovely', 'last_name': 'Spam'})
        writer.writerow({'first_name': 'Wonderful', 'last_name': 'Spam'})

if __name__ == "__main__":
    main()
```